# Naming

Probably one of the hardest but more important things in coding is naming things well.

Some easy to follow rules helps to to maintain consistency and ease of guessing what something is named by following these rules.

- ## Rule 1) Don't use abbreviations

  eg. not StringUtils, use StringUtilities.

  This might seem annoying at first, but there are different ways to abbreviate something, but there is only one way it can be un-abbreviated. For example, abbreviating graphics as gfx, while it is a nice abbreviation, it might equally have been grfx or grphx or one of a dozen other abbreviations. If you didn't already know which, it would otherwise be annoying.

  Also abbreviations can be obvious when you know the context and are deep in to that field, but for others it just makes it harder to gain traction in a new field. For example, a 3x3 matrix being abbreviated to mat3. Code which uses this can add a using statement if they wish, but the canonical name should be un-abbreviated.

- ## Rule 2) Getters, Setters, nouns and verbs

  Getters should be nouns, or prepended with 'Is' or 'Has' words. Setters prepended with 'Set'. Other modifier functions should be verbs. Try to make the function names be descriptive but concise. If it can't be concise, perhaps the function does too many things and needs to be split up in to several smaller functions that are easier to describe by their names.

- ## Rule 3) Capitalization

  First letter of member functions should be capitalized. For acronyms in names, these should be capitalize. Use camel case, eg: IsThingReady(). SetURL(). Use simple hungarian notation - prepend 'a' to parameter names, 'm' to member variables, 's' to statics, 'g' to globals, 'c' to constants, but otherwise local variables have no prefix, but start with a lower case letter. Ideally there are no globals. Enum values don't need to be prepended with 'e' or 'c', but instead are capitalized and use enum classes so they are scoped.